Video-based Automatic Target Recognition

S. Kevin Zhou, Rama Chellappa, Xue Mei, Hao Wu and Qinfen Zheng [*]
Center for Automation Research and Department of ECE
University of Maryland, College Park, MD, 20742
{shaohua,rama,xuemei,wh2003,qinfen}@cfar.umd.edu

Abstract

We present an approach for vehicle classification in IR video sequences by integrating detection, tracking and recognition. The method has two steps. First, the moving target is automatically detected using a detection algorithm. Next, we perform simultaneous tracking and recognition using an appearance-model based particle filter. The tracking result is evaluated at each frame. Low confidence in tracking performance initiates a new cycle of detection, tracking and classification. We demonstrate the robustness of the proposed method using outdoor IR video sequences.

1. Introduction

Recently, video-based vehicle classification has gained much attention, especially in automatic traffic management, surveillance and battlefield awareness. Typically, detection and tracking are often solved before classification. In Lipton et al. (1998), a tracking and classification system is described that can categorize moving objects as vehicles or humans. However, it does not further classify the vehicle into various classes. Wu et al. (2001) uses parameterized model and neural networks for vehicle classification. In Gupte et al. (2002), vehicles are modeled as rectangular patches with certain dynamic behavior and Kalman filtering is used to estimate the vehicle parameters. In Koller (1993), an object classification approach that uses parameterized 3D-models is described. The system uses a 3D polyhedral model to classify vehicles in a traffic sequence. In Kagesawa et al. (2001), a method for recognizing a vehicle's maker and model is proposed. It first creates a compressed database of local features of target vehicles from training images and then matches them with the local features of the probe image for recognition.

In this paper, we tackle the problem of vehicle classification by integrating detection, tracking and recognition. In our system, the moving vehicle is automatically detected, tracked and recognized without any interruptions. The flow chart of our system is shown in Fig.1. The video sequences are input to our system. The moving target is detected using temporal variance analysis. The target is tracked and classified simultaneously using an appearance model and mixtures of probabilistic principal component analysis Tipping and Bishop (1999)(PPCA). Evaluation of the tracking performance is performed at each frame. If the performance falls below some threshold, the cycle of detection, tracking and classification is re-initiated, otherwise the tracking and classification propagates to the next frame.

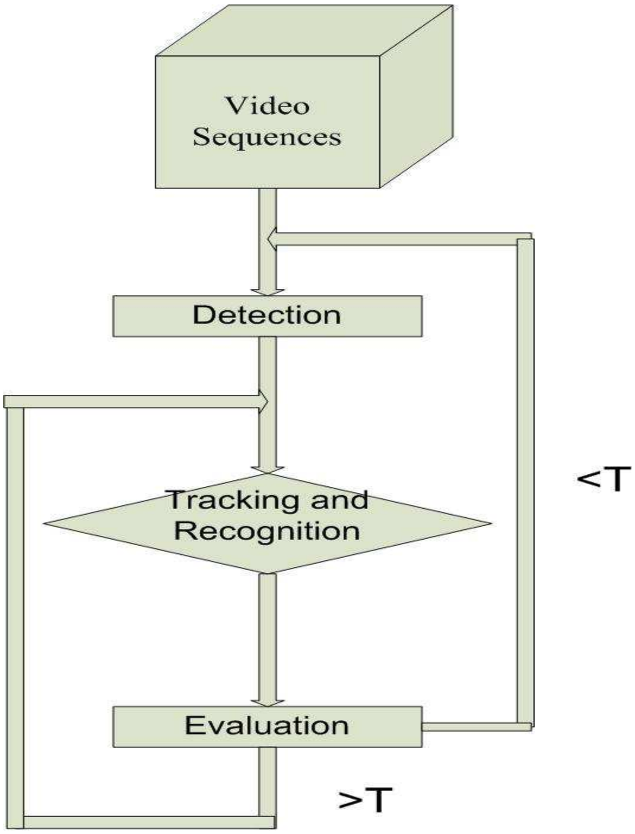


Figure 1: A flow chart of our system.

There are four types of vehicles used in the experiment.

[*] Prepared through collaborative participation in the Advanced/Sensors Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-02-0008

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **00 DEC 2004**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Video-based Automatic Target Recognition**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Center for Automation Research and Department of ECE University of Maryland, College Park, MD, 20742**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

See also ADM001736, Proceedings for the Army Science Conference (24th) Held on 29 November - 2 December 2004 in Orlando, Florida., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**6**	

They are ‘m60’, ‘brdm’, ‘wetting’ and ‘bmp’. Four probe video sequences each of which contains different vehicles are used for classification. Fig.2 shows two image samples from the probe video sequence ‘bmp1’. Fig.2(a) shows the side view of ‘bmp’ and (b) the frontal view. The target-to-background contrast is very low for the IR images. This adds much difficulty for the detection and tracking of moving target.

Unlike Zhou et al. (2003)’s method which manually selects the moving target in the first frame, we automatically select it using a detection algorithm. Because of the presence of smoke and dust in IR videos as showed in Fig.2, it is hard to position a tight rectangular bounding box from the detection algorithm. Consequently, the tracker drifts quickly. This brings a need for the evaluation of the tracking performance. The evaluation generates a confidence measure to indicate whether we should restart the detection once the tracking confidence falls below a threshold.

We use mixtures PPCA Tipping and Bishop (1999) for appearance modeling. We then compute the posteriori probability of finding the appearance of each object in the given video and assign the label corresponding to the maximum.

The rest of this paper is organized as follows. Section 2 describes the detection algorithm. Section 3 describes the tracking and classification algorithm. Section 4 details the simultaneous evaluation for the tracking and section 5 describes experiments. Finally, conclusion and future work are discussed in section 6.

2. Target Detection

Detection plays an important role in our system. It is a prerequisite for the tracking. It gives an initial bounding box surrounding the target and re-initialize the target if tracking confidence measure is low.

Given a video sequences $\{I_i\}$, we set $m_1 = I_1$ and $mv_1 = I_1 \times I_1$. The operator $\times$ is the element-by-element produce of two matrices. The following m_i, mv_i and $imvar_i$ are defined as

$$m_i = \{(N-1) * m_{i-1} + I_i\}/N, \tag{1}$$

$$mv_i = \{(N-1) * mv_{i-1} + I_i \times I_i\}/N, \tag{2}$$

$$imvar_i = \sqrt{mv_i - m_i \times m_i}, \tag{3}$$

where N is the window size for detection which is 150 in our experiment.

For the element $p(i,j)$ in $imvar_i$, we will set $p(i,j) = 1$ if $p(i,j) > T$, otherwise $p(i,j) = 0$, where T is the threshold. Now $imvar_i$ is converted to a binary image which we call the variance image. We then select the rectangular bounding box for the moving target by checking $p(i,j) = 1$ in the image.

Figures 3 and 4 show the detection results for ‘brdm’ and ‘m60’ respectively. From Fig.3, we can see the bounding box is very big due to the smoke emitted by the vehicle. In Fig.4, the similarity between the environment and target affect the bounding box localization.

3. Target Tracking and Classification

This section describes the vehicle tracking and classification algorithm. In section 2.1, the state space model used for tracking and classification is described. Tracking and classification are implemented simultaneously by estimating the posterior distribution . In section 2.2, the mixtures of PPCA algorithm for estimating the distribution of identity variable for the classification is detailed.

3.1. State Space Model

A time series state space model uses the state variable $x_t = \{n_t, \theta_t\}$, which includes identity variable n_t and 2D affine transformation motion parameters θ_t. The system equation is written as

$$n_t = n_{t-1} \qquad \theta_t = \theta_{t-1} + u_t,\ t \geq 1 \tag{4}$$

where we assume that the motion variable follows a Markov process with u_t as a white Gaussian noise process. $n_t \in N = \{1, 2, \cdots, N\}$ indexes the gallery set $\{I_1, I_2, \cdots, I_N\}$.

A simple formulation of the observation equation can be characterized as

$$Z_t = T\{Y_t; \theta_t\} = I_{n_t} + V_t \tag{5}$$

Where Z_t is the image patch of interest in the video frame, T is an affine transformation to normalize the image to the same size of the gallery images, and V_t is the noise. The observation equation is equivalently characterized by the likelihood $p(Y_t|n_t, \theta_t) = p(Z_t|n_t)$. In the next section, we define $p(Z_t|n_t)$ as mixtures of PPCA.

The essence of the approach is posterior probability computation, i.e. computing $p(n_t, \theta_t|Y_{1:t})$, whose marginal posterior probability $p(n_t|Y_{1:t})$ solves the classification task and marginal posterior probability $p(\theta_t|Y_{1:t})$ solves the tracking task.

Classification is based on a Maximum A Posteriori (MAP) decision rule, namely finding n_t that maximizes $p(n_t|Y_{1:t})$. The Sequential Importance Sampling(SIS) Liu and Chen (1998) method is used to approximate and propagate the posterior probability $p(n_t, \theta_t|Y_{1:t})$, and marginalization over variable θ_t is carried out before applying the classification rule. Detailed descriptions can be found in Zhou et al. (2004).

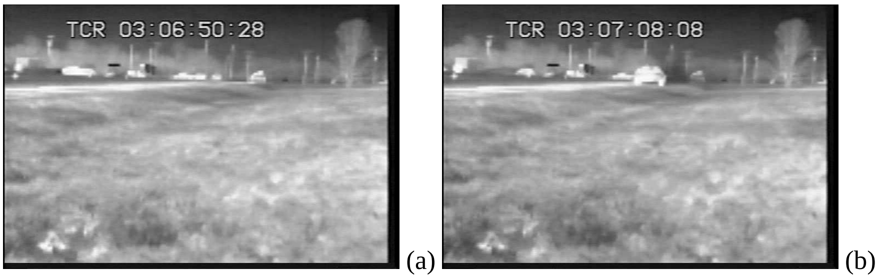


Figure 2: Image frames from video sequences ‘bmp1’. It shows the side view(a) and the frontal view(b) of the vehicle.

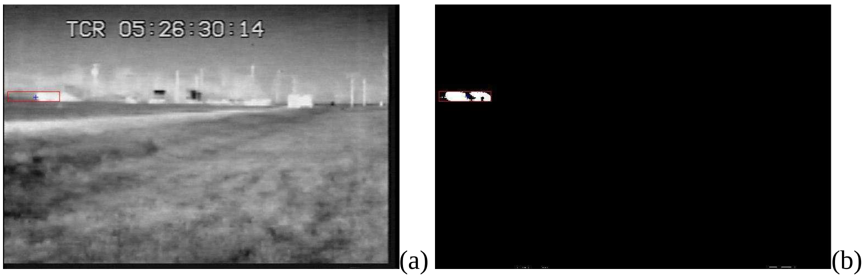


Figure 3: Detection result for ‘brdm’. (a) is the original image and (b) is the detected target chip. It shows how the smoke emitted by the vehicle affects the detection result.

3.2 Mixtures of Probabilistic PCA

Subspace analysis techniques have attracted growing interest in computer vision research. In particular, eigenvector decomposition has been shown to be an effective tool for solving problems by using a low-dimensional vector to represent high-dimensional vector. Here we will follow Tipping and Bishop (1999) for the mixtures of PPCA.

Given a set of m by n images $\{Z_i\}$, we form a set of vectors $\{t_i\}$, where $t_i \in R^{d=mn}$, by lexicographic ordering of the pixel elements of each image Z_i. For any t in $\{t_i\}$, we relate it to a corresponding γ-dimensional vector variable x as:

$$t = Wx + \mu + \varepsilon \tag{6}$$

where $d \gg \gamma$ and μ is the mean of the x.

For the case of isotropic noise $\varepsilon \sim N(0, \sigma^2 I)$, the distribution over t-space for a given x of the form

$$p(t|x) = (2\pi\sigma^2)^{-d/2} exp\{-\frac{1}{2\sigma^2}||t - Wx - \mu||^2\} \tag{7}$$

With a Gaussian prior for the x, we obtain the marginal distribution of t

$$p(t) = (2\pi)^{-d/2}|C|^{-1/2} exp\{-\frac{1}{2}(t-\mu)^T C^{-1}(t-\mu)\}, \tag{8}$$

where the covariance is

$$C = \sigma^2 I + WW^T. \tag{9}$$

The mixtures of PPCA can model more complex data structures. The model parameters are determined using maximum likelihood estimation. The mixture model is defined as:

$$p(t) = \sum_{i=1}^{M} \pi_i p(t|i) \tag{10}$$

where $p(t|i)$ is a single PPCA model and π_i is the corresponding mixing proportion, with $\pi_i \geq 0$ and $\sum \pi_i = 1$. Now the three parameters μ, W and σ^2 are associated with each of the M mixture components. We use an iterative EM algorithm for estimating the model parameters.

4 Tracking Evaluation

Most practical tracking systems often fail under some situations. This could be either because of illumination changes, pose variation or occlusion. Therefore, the need for automatic performance evaluation emerges in these applications. Fig.5 shows the tracking result after running the tracker for some time. The bounding box is so large that one concludes that the tracker is already failing. Hence, evaluation is necessary to help us terminate tracking and restart the detection-tracking-classification cycle.

Our evaluation algorithm is based on measuring the appearance similarity and tracking uncertainty. The following features are examined in our evaluation:

1. Trace complexity q_{tc}: We define the trace complexity

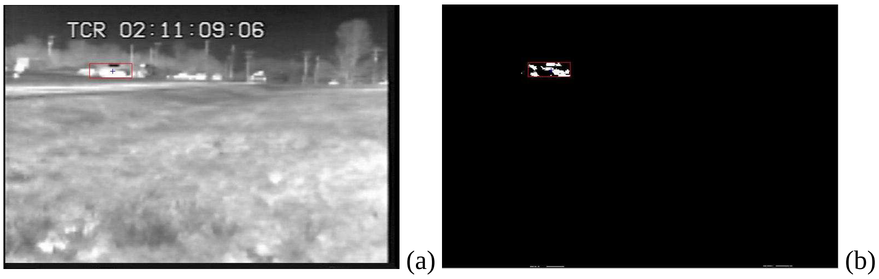


Figure 4: Detection result for ‘m60’. (a) is the original image and (b) is the detected target chip. It shows how contrast and SNR affect the detection result.

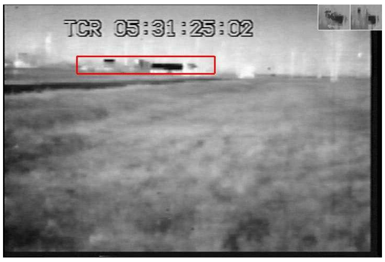


Figure 5: An example of poor tracking.

as the ratio of the curve length and straight length between the target centroids in different frames.

2. Motion step q_{ms}: It is defined as the distance between the box centers in two consecutive frames.

3. Scale change q_{sc}: To examine changes in object scale, we use two clues. One is the ratio of the current area to the initial area, the other is the scale change velocity.

4. Shape similarity q_{ss}: The change in the aspect ratio of the bounding box is also useful in providing some information about the object shape. It is defined as the ratio of the current aspect ratio over the initial ratio.

5. Appearance change q_{ac}: Three measures are used in our algorithm, the first one is the absolute pixel by pixel change between the current frame and the initial frame, the second one is the histogram difference between the current frame and the initial frame and the last one is related to the tracking algorithm over which the proposed algorithm was tested.

To obtain a comprehensive measure of the tracking performance, we combine the above five indicators. We first use empirical thresholds to find whether the tracker is uncertain according to the above five metrics, then we sum the five indicators using different weights to arrive at a confidence measure q. If the sum drops below some threshold, we conclude that the tracking performance is poor and needs re-initialization.

$$q = \sum_{j \in J} w_j I[q_j < \lambda_j], \;\; J \in \{tc, ms, sc, ss, ac\} \qquad (11)$$

where w_j and λ_j are the corresponding weights and thresholds for the evaluation.

5 Experiments

In this section, we give details of our implementation. Training and testing are described in the next two sections respectively. In our experiment, the vehicle motion is characterized by $\theta = (a_1, a_2, a_3, a_4, t_x, t_y)$, where $\{a_1, a_2, a_3, a_4\}$ are the deformation parameters and (t_x, t_y) are the 2D translation parameters. By applying an affine transformation using θ as parameters, we crop the region of interest so that it has the same size as the still template in the gallery and perform zero-mean-unit-variance normalization. The region of interest is 24×30 in size.

5.1 Training

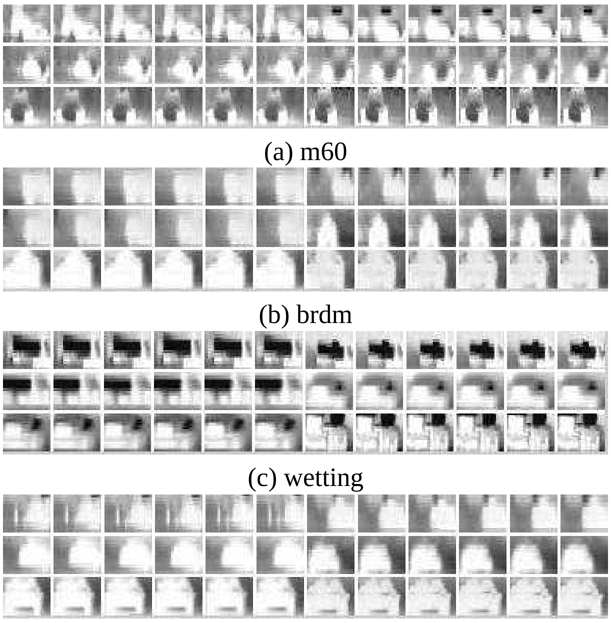
We use one video sequence for each vehicle and obtain the tracking result. Then we select 36 images for each vehicle in the gallery. The pertinent parameters for the experiment are $M = 2$ and $\gamma = 15$.

Fig.6 is the gallery of the vehicle images. There are a total of 144 images in the gallery. They are ‘m60’, ‘brdm’, ‘wetting’ and ‘bmp’ from top to bottom, each has three rows.

After we have the gallery images, we use mixtures of PPCA to estimate the parameters π_i,μ_i, W_i and σ_i^2.

5.2 Testing

For each frame, we get the motion parameters after tracking and cropping out the region of interest of size 24×30 from the original image. After performing zero mean and unit variance operation, we substitute the vector as t into equation (10) and get the probabilities for each vehicle. We



(a) m60

(b) brdm

(c) wetting

(d) bmp

Figure 6: Gallery of vehicle images. The image size is 24 by 30.

pick the vehicle which has the highest probability as our classification result after normalization. The probabilities propagate to the next frame. In each frame, if the confidence measure is below some threshold, the detection will restart 20 frames before the drifting point and tracking and classification will restart too.

Fig.7 shows the tracking and recognition results for 'wetting1' and Fig.8 is for 'bmp1'. In Fig.7(a), the image on the top is the tracking result for the current frame. We put a bounding box for the vehicle which we are tracking in each frame with a different color for different vehicles. The image on the left of the bottom is the classification score which is the probability of seeing each vehicle in the video. It shows the result from the first frame to the current frame. The image to the right is the tracking confidence measure which represents the probability of the correct tracking result. We will restart detection and tracking if the measure falls below the threshold of 0.5. The same description applies to Fig7(b) and Fig.8.

From Fig.7, we observe that the recognition result for the 'wetting1' is very good because a high probability is associated with 'wetting' (dotted blue line) on almost every frame. There are several peaks and valleys for the dotted blue line due to the re-initialization of the tracking and the evaluation probability on the right drops very quickly at corresponding frames. In Fig.8, for the recognition of 'bmp1', it is confused by 'brdm' for the first half of the sequence. It is very hard to get an initial tight bounding box due to the smoke emitted by 'bmp1' using the detection algorithm. The tracker quickly drifts away after about 40 frames given the initial location. For frame 99, the result is incorrect, as it gives 'brdm' as the recognition result. The result becomes stable and correct after 400 frames. After running the whole video sequence, the correct recognition result is quite good. For this situation, we will classify that the vehicle we are tracking is 'bmp' which yields the correct result.

	m60	brdm	wetting	bmp
m60	93.82%	3.17%	0	3.01%
brdm	0	85.64%	0	14.36%
wetting	0	0	95.65%	4.35%
bmp	0	18.85%	0	81.15%

Table 1: Confusion matrix for vehicle classification experiment.

The results of the experiment are summarized in Table 1. Each number in a row is the recognition percentage of the vehicle. Taking the second row as an example, 93.82% of the whole sequence recognizes the vehicle as 'bmp', while 3.17% as 'brdm' and 3.01% as 'bmp'. No frame recognizes it as 'wetting'. The elements in the diagonal give the correct recognition score for our experiment. The overall accuracy of the recognition is 89.07%.

6 Conclusion and Future Work

In this paper, we have proposed an approach for vehicle classification by integrating detection, tracking and recognition. The experiment results prove our method's robustness and effectiveness.

Our future work will include improving detection, tracking and evaluation algorithms and developing a more robust and stable recognition algorithm. Large data set will also be tested to obtain a more general analysis.

References

Gupte, S., Masoud, O., Martin, R., and Papanikolopoulos, N. (2002). Detection and classification of vehicles. *IEEE Transactions on Intelligent Transportation Systems*.

Kagesawa, M., Ueno, S., Ikeuchi, K., and Kashiwagi, H. (2001). Recognizing vehicles in infrared images using imap parallel vision board. *Intelligent Transportation Systems, IEEE Transactions on*, 2:10–17.

Koller, D. (1993). Moving object recognition and classification based on recursive shape parameter estimation. *12th Israel Conference on Artificial Intelligence, Computer Vision*.

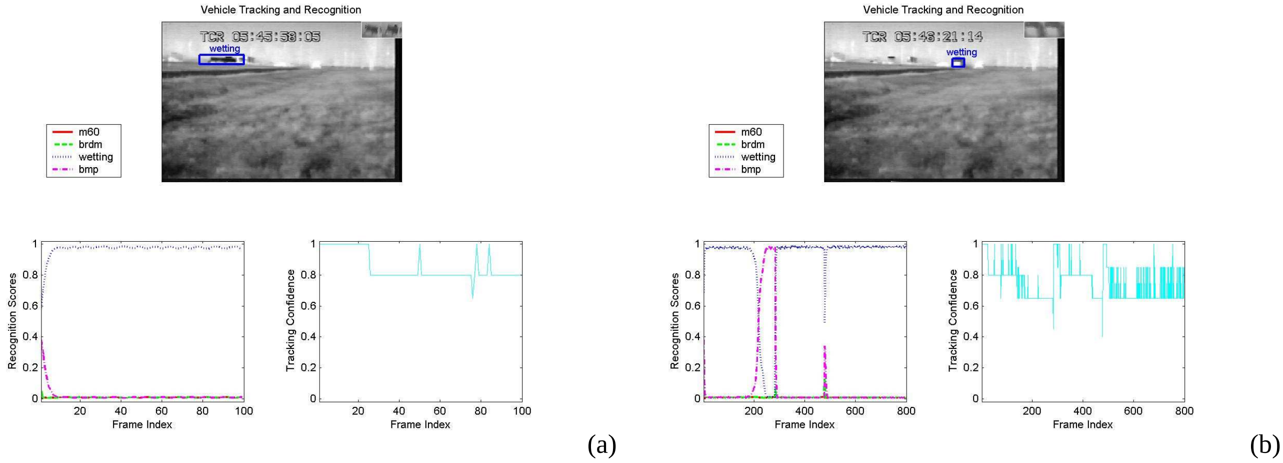


Figure 7: Tracking and recognition results for ‘wetting1’. The results are from frame 1 to 99 for (a) and frame 1 to 799 for (b). The top panel shows the original image and tracking result, the bottom left panel shows the recognition density $p(n_t|Y_{1:t})$, and the bottom right panel shows the tracking confidence q.

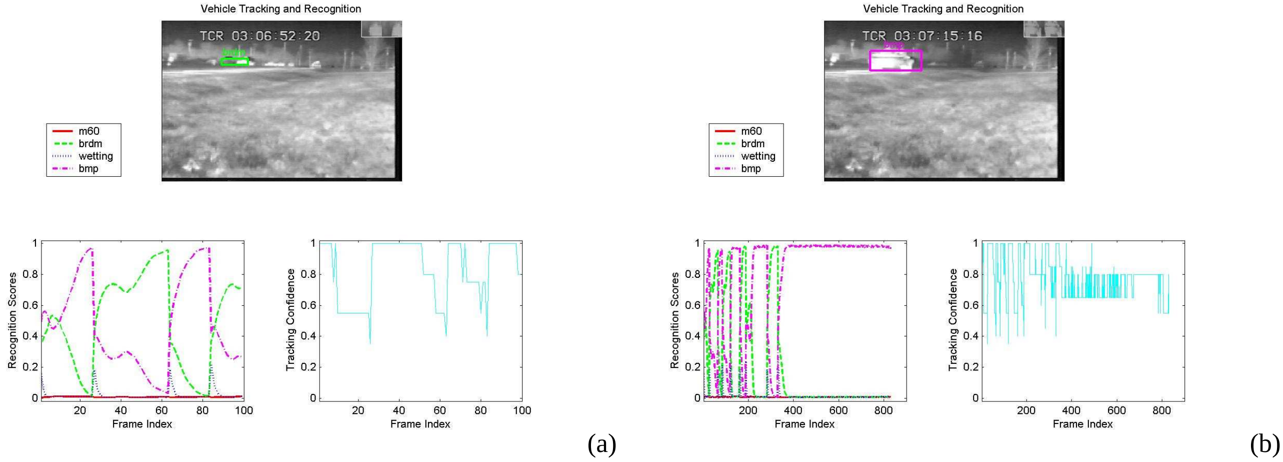


Figure 8: Tracking and recognition results for ‘bmp1’. The results are from frame 1 to 99 for (a) and frame 1 to 830 for (b).

Lipton, A., Fujiyoshi, H., and Patil, R. (1998). Moving target classification and tracking from real-time video. *Proc. Of the Image Understanding Workshop*.

Liu, J. and Chen, R. (1998). Sequential monte carlo for dynamic systems. *Journal of the American Statistical Association*, 93:1031C1041.

Tipping, M. and Bishop, C. (1999). Mixtures of probabilistic principal component analysers. *Neural Computing*, 11:443–482.

Wu, W., Zhang, Q., and Wang, M. (2001). A method of vehicle classification using models and neural networks. *Vehicular Technology Conference*.

Zhou, S., Chellappa, R., and Moghaddam, B. (2004). Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*.

Zhou, S., Krueger, V., and Chellappa, R. (2003). Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, 91:214–245.